Active Directory Abuse Primitive and Operation Security

Mars Cheng and Dexter Chen

PSIRT and Threat Research, TXOne Networks Inc.

April 21, 2023 @ HITB 2023 Amsterdam

Mars Cheng and Dexter Chen

Mars Cheng

Dexter Chen

Threat Research Manager, PSIRT and Threat Research at TXOne Networks

- Executive Director, Association of Hackers in Taiwan
- ICS/SCADA, IoT, Malware Analysis, and Enterprise Security
- Spoke at Black Hat USA and Europe, RSA Conference, DEF CON, HITCON, FIRST, SecTor, HITB, SINCON, ICS Cyber Security Conference USA and Asia, CYBERSEC, and InfoSec Taiwan.
- Instructor of HITCON Training 2022/2021/2020/2019, CCoE Taiwan, Ministry of Education, Ministry of National Defense, Ministry of Economic Affairs in Taiwan, and Listed companies
- General Coordinator of HITCON (Hacks In Taiwan Conference) PEACE 2022 and 2021
- 10+ CVEs and 3 SCI Journals
- Member of CHROOT Study Group

Threat Researcher, PSIRT and Threat Research at TXOne Networks

- Supervisor, Association of Hackers in Taiwan
- Red Teaming, Active Directory Security, and Penetration Testing
- Spoke at Black Hat MEA, CODE BLUE, HITCON, CYBERSEC
- Instructor of HITCON Training 2022/2021/2020, CCoE Taiwan, and Ministry of National Defense
- OSWE and OSCP holder

Outline

01 | Active Directory (AD) Overview and What is Operation Security (OPSEC)?

02 | 4 Attack Paths Demonstration

03 | Takeaways

Active Directory (AD) Overview and What is Operation Security (OPSEC)?

txOne networks

Active Directory

- Active Directory (AD) is a directory service developed by Microsoft for Centralized Domain Management Initially
  - Nowadays, Active Directory became an umbrella title for a broad range of directory-based identity-related services
- Key features in Active Directory: Lightweight Directory Access Protocol (LDAP) versions 2/3, Microsoft's version of Kerberos, and DNS
- Many companies use Active Directory for Single Sign-On(SSO), which allows internal services such as site, email access and other servers to authenticate users is based on AD

Blue Team:

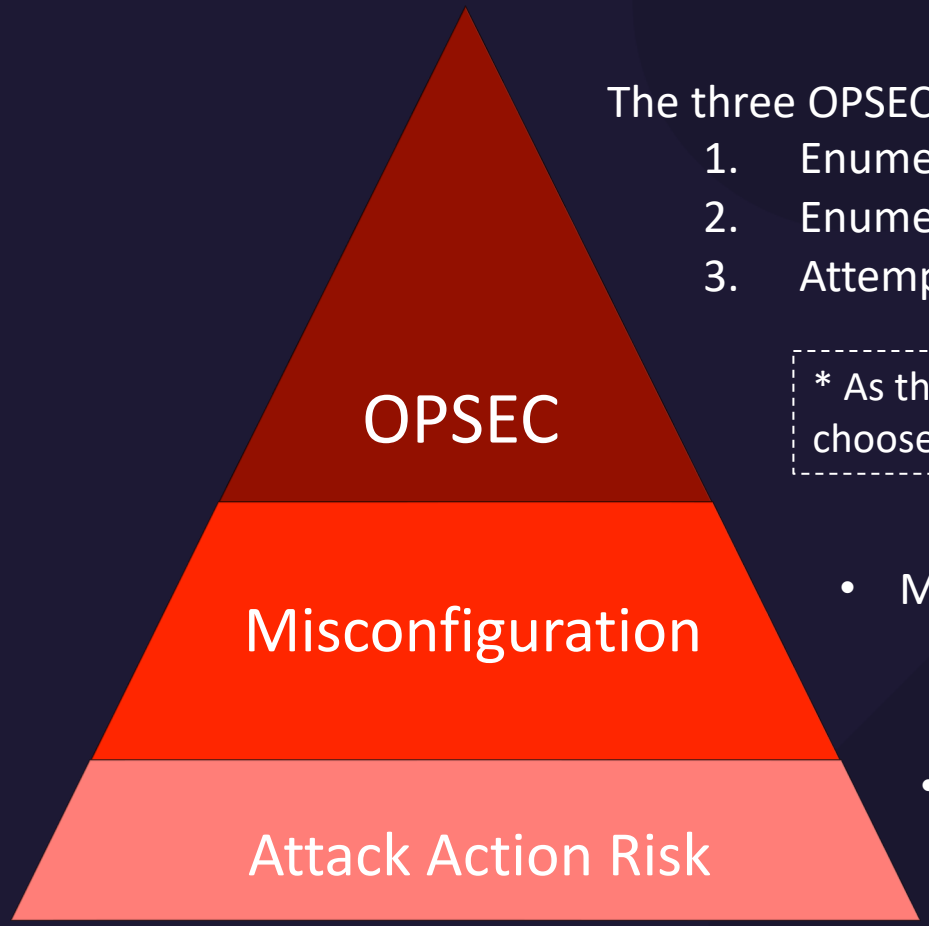
We are aware of the importance of Active Directory and try to take some corresponding actions to detect and prevent AD attacks

Red Team:

We have Operation Security (OPSEC)

“Determine if friendly actions can be observed by enemy intelligence that could compromise the operation”

OPSEC Approach for Active Directory Attacks



The three OPSEC steps for AD can be:

1. Enumerate the security posture of the target
2. Enumerate possible detection indicators for tradecraft used
3. Attempt to make these indicators align the baseline for detection prevention

* As there can be multiple methods for a specific abuse primitive, we can always choose the one that has fewer indicators for detection

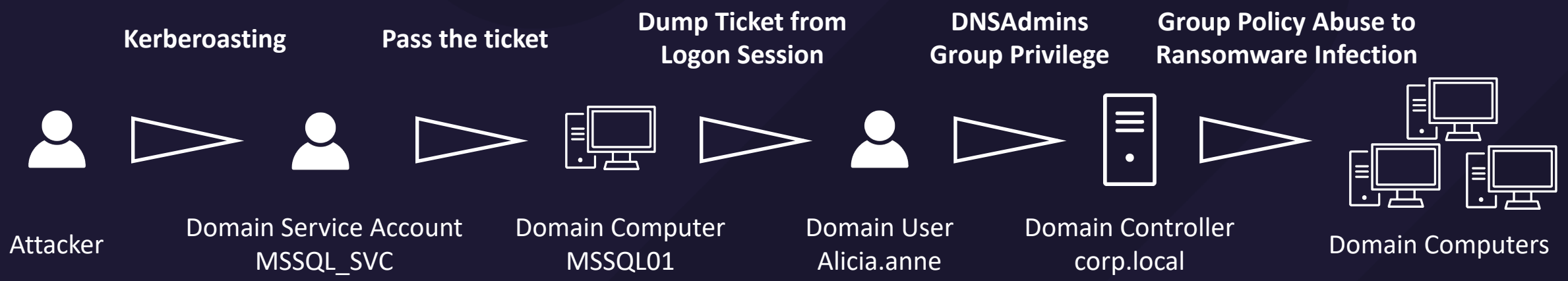
- Most Attacks in Active Directory are in a form of abuse primitive
  - The detection is heavily based on the behavior

- Attack Action performed must be considered for the risk of being detected
  - Once the Blue Team is alerted, the campaign is mostly burn
  - Indicators should be avoided as must as possible

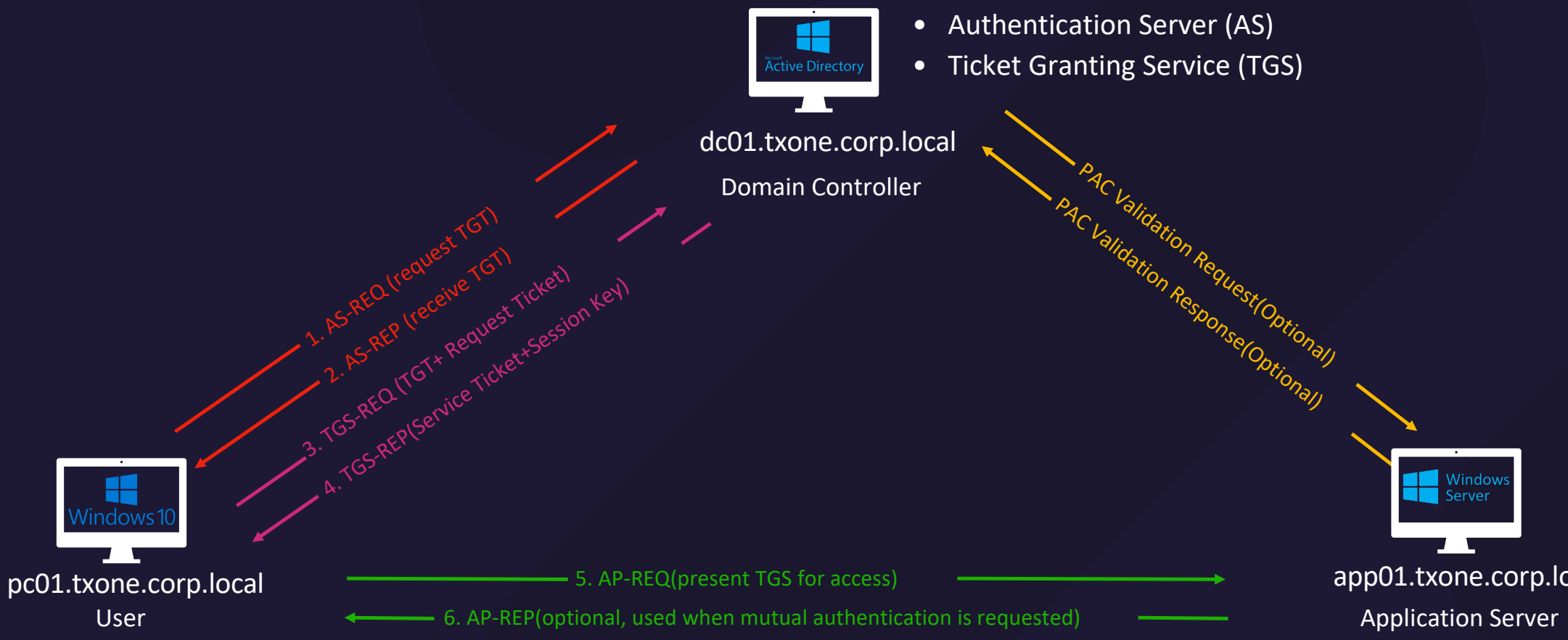
4 Attack Paths Demonstration

Attack Path I Overview

From Kerberoasting to Group Policy Abuse for Ransomware Infection



Kerberos Authentication Overview



Attack Path Preview - Kerberoasting

- When we want to use a service, the service ticket is requested first and presented to the service for access (TGS-REQ/REP)
- The portion of the service ticket is encrypted by **service account password hash** as the key that is prone to **brute force** attack
  - The attacker brute force the password by attempting to decrypt the encryption part
  - Service tickets that are abused in kerberoasting are often in RC4 encryption type

- Attack Procedures
  1. Enumerate user account with SPN set
  2. Perform TGS request for the target service ticket
  3. Brute Force the encryption part of the ticket for the service account password
     - This step is done in the attacker's local machine and will not be considered for OPSEC

Attack Path Preview - Pass the Ticket

- Attackers use stolen/forged Kerberos tickets to authenticate as a domain account and access the system
- Attack Procedure
  - Obtain a ticket through various means
    - Use stolen password hash
    - Forged Golden(TGT)/Silver(Service) ticket
    - Dumped TGT/Service ticket
  - (Windows) Import this obtained TGT/Service Ticket into the logon session to impersonate the target user and access a service
  - (Linux) Directly supply obtained TGT/Service Ticket to impersonate the target user and access a service

DNS Server Management Protocol

- DNS Server Management Protocol is on top of Microsoft Remote Procedure Call (RPC), and contains methods to **perform DNS-related operation**
  - UUID is 50ABC2A4–574D- 40B3–9D66-EE4FD5FBA076
- One of the DNS operations, the ServerLevelPluginDll, allows the DNS to have a custom plug-in using a function
  - For resolving name queries that are outside the scope of all locally hosted zones
    e.g. non-domain joined Linux machine

Methods in RPC Opnum Order

Method	Description
R_DnssrvOperation	Invokes a specified set of server functions. This method is obsoleted by R_DnssrvOperation2 (Opnum 5) (section 3.1.4.6). Opnum: 0

3.1.4.1 R_DnssrvOperation (Opnum 0)

Article • 08/11/2021 • 79 minutes to read Feedback

The R_DnssrvOperation method is used to invoke a set of server functions specified by pszOperation.

```
LONG R_DnssrvOperation(
  [in]                          handle_t            hBindingHandle,
  [in, unique, string]          LPCWSTR             pwszServerName,
  [in, unique, string]          LPCSTR              pszZone,
  [in]                          DWORD               dwContext,
  [in, unique, string]          LPCSTR              pszOperation,
  [in]                          DWORD               dwTypeId,
  [in, switch_is(dwTypeId)]     DNSSRV_RPC_UNION    pData
);
```

Attack Path Preview - DNSAdmins Group Privilege

- When the ServerLevelPluginDll feature is configured, the dll path value is stored in the registry key
  - KEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DNS\Parameters\ServerLevelPluginDll
  - the DLL must contain the three functions DnsPluginInitialize, DnsPluginCleanup, and DnsPluginQuery. Otherwise, the loading will fail
  - The return integer value must be 0 to indicate the success of function execution
- Only after the restart of the DNS service, will the DLL specified in the registry will then be loaded
- If the DLL fails to load for any reason, the DNS service **will fail to start**

Attack Path Preview - Group Policy Abuse

- Group Policy has a huge number of settings to manipulate, giving you several opportunities to compromise machines and users
- Possible scenarios you would consider abusing GPO
  - You compromised a user that has **edit right over a GPO**
  - You have Domain Admin(DA) access in the environment, using **GPO for persistence**
- Overall, you are **manipulating the setting for some malicious purpose**
  - Add the user to the local administrators group
  - Add new computer startup script
  - Edit user logon script

Operation Security - Kerberoasting

- Before service ticket is requested, attacker will **enumerate the users with SPN set via LDAP**
- Possible Detection Indicator
  - LDAP query that specifically asks for user accounts with SPN set
- OPSEC Actions
  - Search LDAP entire users and filter them manually for having SPN set
  - Request specific target service target later

```
Get-DomainUser | ?{$_.serviceprincipal -ne $null}
```

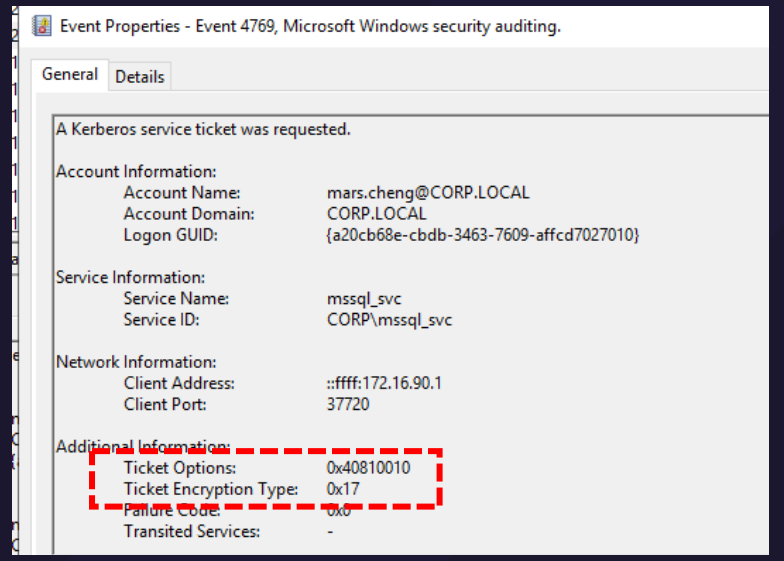
```
<Data Name="ScopeOfSearch">2</Data>
<Data Name="SearchFilter">(&(samAccountType=805306368))</Data>
<Data Name="DistinguishedName">DC=CORP,DC=LOCAL</Data>
<Data Name="AttributeList" />
<Data Name="ProcessId">0x14F4</Data>
```

When query for entire users, LDAP query will **not contain** the detection indicator

LDAP query that specifically asks for user accounts **with SPN set, will be Possible Detection Indicator**

Operation Security - Kerberoasting

- After a service account is targeted, the attacker performs TGS request to obtain the service ticket for brute force
- Possible Detection Indicator
  - The service Ticket field could be baselined for detection
    1. Ticket Options = 0x40810010 (used by tools such as Rubeus)
       - The baseline value is 0x40810000 by default
    2. Encryption Type = 0x17 (RC4 Encryption for brute force)
       - The baseline value can be 0x12 if it is enabled



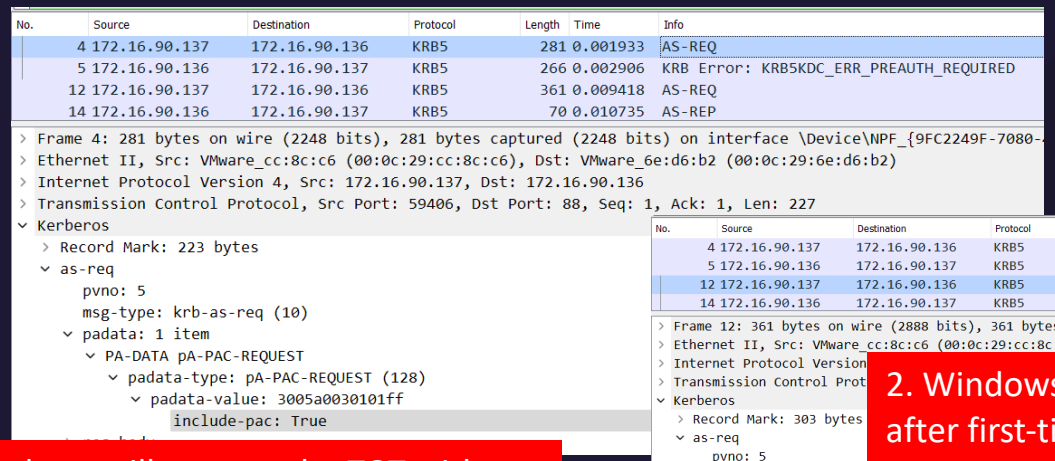
- OPSEC Actions
  - Enumerate encryption type supported by target account and baseline of ticket options in the target environment
    - Tuning the tradecraft respectively
      - e.g., use /enctype flag in Rubeus to specify encryption type to AES256 if target account supports it
      - Use Orpheus, written by TrustedSec, to manually configure the ticket options in TGS request to 0x40810000

Operation Security - Pass the Ticket

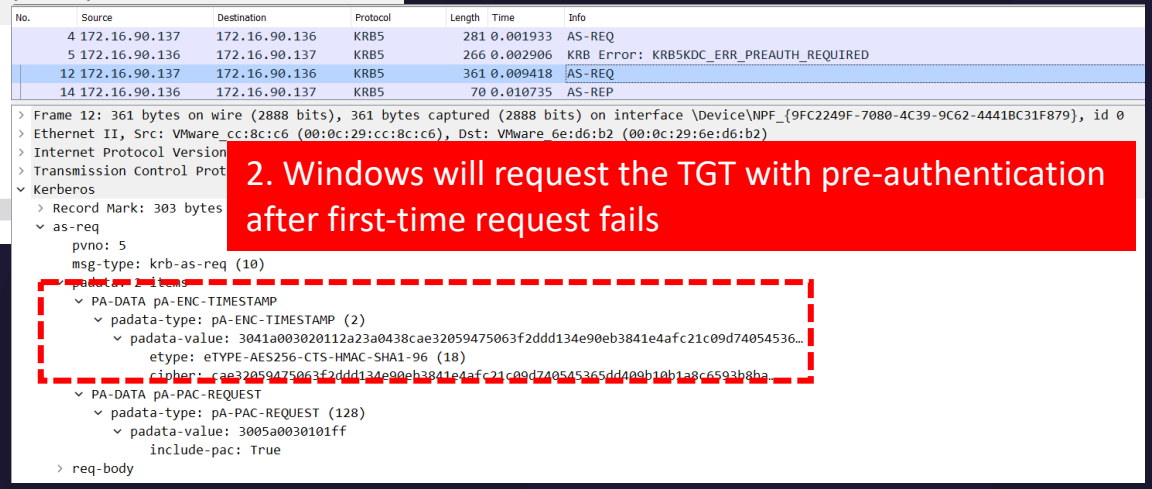
- Attackers use stolen password hash to request Kerberos ticket and impersonate the target user and access a service
- Possible Detection Indicator
  - Send AS-REQ with pre-authentication directly
    - There is no PREAUTH_REQUIRED error

Source	Destination	Protocol	Length	Info
172.16.90.137	172.16.90.136	LDAP	408	bindRequest(3) "<ROOT>" sasl
172.16.90.136	172.16.90.137	LDAP	265	bindResponse(3) success
172.16.90.137	172.16.90.136	KRB5	279	AS-REQ
172.16.90.136	172.16.90.137	KRB5	1414	AS-REP

Rubeus ask TGT with
pre-authentication directly by default



1. Windows will request the TGT without pre-authentication first by default



2. Windows will request the TGT with pre-authentication after first-time request fails

Operation Security - Pass the Ticket

- OPSEC Actions
  - We can use /opsec handle in Rubeus to request a TGT that mimic windows behavior
    - Send AS-REQ requests without pre-authentication first and with pre-authentication after first-time request fails

```
PS C:\Users\mars.cheng\Desktop> .\Rubeus.exe asktgt /user:mars.cheng /password:Password! /domain:corp.local /enctype:aes256 /opsec

  ______        _
  (_____ \      | |
   _____) )_   _| |__  _____ _   _  ___
  |  __  /| | | |  _ \| ___ | | | |/___)
  | |  \ \| |_| | |_) ) ____| |_| |___ |
  |_|   |_|____/|____/|_____)____/(___/

  v2.2.2

[*] Action: Ask TGT

[*] Using domain controller: DC01.corp.local (172.16.90.136)
[!] Pre-Authentication required!
[!]     AES256 Salt: CORP.LOCALmars.cheng
[*] Using aes256_cts_hmac_sha1 hash: 3D75B8F477A8B9D0EC040539ECE88C01C51BBC4F7CCBDC9B5A1B43AD9C42C292
[*] Building AS-REQ (w/ preauth) for: 'corp.local\mars.cheng'
[*] Using domain controller: 172.16.90.136:88
[+] TGT request successful!
[*] base64(ticket.kirbi):
```

kerberos

No.	Source	Destination	Protocol	Length	Time	Info
490	172.16.90.137	172.16.90.136	KRB5	272	213.3646…	AS-REQ[BoundErrorUnreassembled Packet]
491	172.16.90.136	172.16.90.137	KRB5	242	213.3662…	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
511	172.16.90.137	172.16.90.136	KRB5	283	216.0208…	AS-REQ
512	172.16.90.136	172.16.90.137	KRB5	1508	216.0230…	AS-REP

```
> Frame 490: 272 bytes on wire (2176 bits), 272 bytes captured (2176 bits) on interface \Device\NPF_{9FC2249F-7080-4C39-9C62-4441BC31F879}, id 0
> Ethernet II, Src: VMware_cc:8c:c6 (00:0c:29:cc:8c:c6), Dst: VMware_6e:d6:b2 (00:0c:29:6e:d6:b2)
> Internet Protocol Version 4, Src: 172.16.90.137, Dst: 172.16.90.136
> Transmission Control Protocol, Src Port: 60312, Dst Port: 88, Seq: 5, Ack: 1, Len: 218
> [2 Reassembled TCP Segments (222 bytes): #488(4), #490(218)]
v Kerberos
  > Record Mark: 218 bytes
  v as-req
      pvno: 5
      msg-type: krb-as-req (10)
    v padata: 1 item
      v PA-DATA pA-PAC-REQUEST
        v padata-type: pA-PAC-REQUEST (128)
          v padata-value: 3005a003010101
              include-pac: True
    > req-body
```

With /opsec handle, Rubeus will request the TGT without pre-authentication first

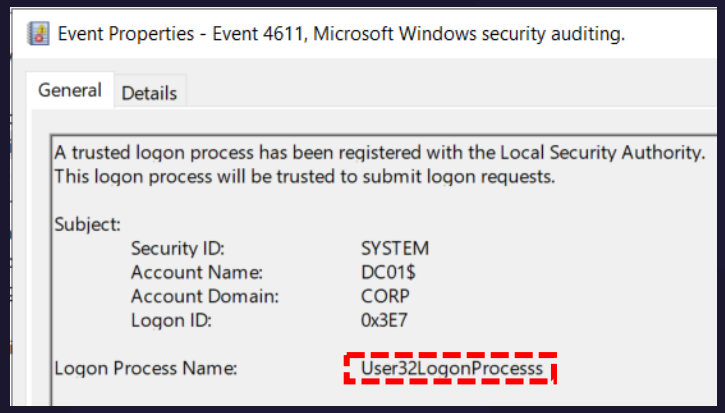
Operation Security - Pass the Ticket

- In Windows, attacker **imports the obtained TGT/Service Ticket** into the logon session to impersonate the target user and access a service
- Possible Detection Indicator
  - Imported TGT/Service Ticket username is different from logon session username
- OPSEC Actions
  - After attacks actions for target users are done, purging the imported TGT/Service Ticket to clean up the trace
    - e.g., “klist.exe purge”

Operation Security - Dump Ticket

- Attacker **registered a logon process** name to get an LSA handle and dump the TGT/Service Ticket from logon session in a machine
- Possible Detection Indicator
  - Rubeus intentionally put the logon process as “User32LogonProcesss” with three s in the end
- OPSEC Actions
  - Modify the code in Rubeus, removing the extra “s”, to make logon process name looks normal

```
57          public static IntPtr LsaRegisterLogonProcessHelper()
58          {
59              // helper that establishes a connection to the LSA server and verifies that the
60              //  used for Kerberos ticket enumeration for ALL users
61
62              var logonProcessName = "User32LogonProcesss"; // yes I know this is "weird" ;)
63              Interop.LSA_STRING_IN LSAString;
64              var lsaHandle = IntPtr.Zero;
65              UInt64 securityMode = 0;
```

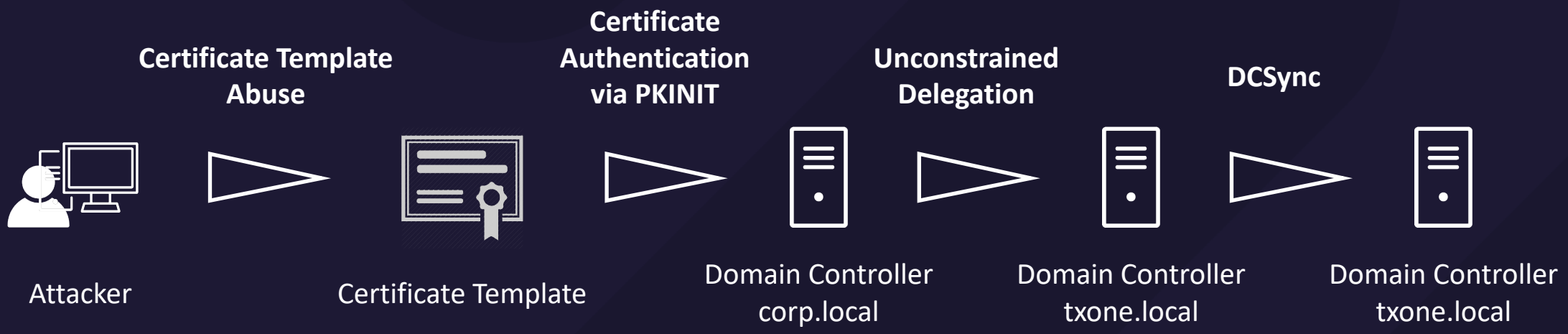


Operation Security - DNSAdmins Abuse

- DNSAdmins abuse primitive is specifically required to config ServerLevelPluginDll for command execution
- Possible Detection Indicator
  - Registry key “HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\services\DNS\Parameters\ServerLevelPluginDll” **is populated** after being abused for command execution on DC
- OPSEC Actions
  - Remove the configuration from the registry key when the abuse is done might not leave a trace for the defender
  - Members of DNSAdmins have full control of DNS Service, it can also be leveraged for NTLM relay by modifying the existing DNS record with DNSAdmins permission
    - e.g., change the DC record to attacker compromised machine

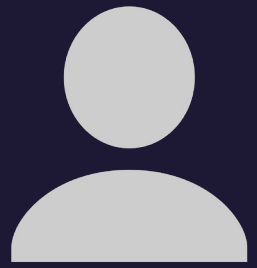
Attack Path II Overview

Abuse AD CS to Compromise Entire Domain



Attack Path Preview - Certificate Template Abuse

- Vulnerable template allow attacker to abuse it to obtain a privileged user certificate that can be used for Kerberos authentication (PKINIT)



User

1. Client generates public/private key pair

2. Client sends a certificate request(CSR) to Enterprise CA Server

Certificate Request(CSR)

CSR
Template:CodeSign
Subject: Corp/user
EKU: Code Signing
…
Public Key: <public key>

CodeSign Template
EKU: Code Signing
Enrollment Permissions: Domain Users
Request Approval? No

Active Directory Certificagte Service

Enterprise CA

3. Check the following requirements

- Does the certificate template in the enable certificate templates?
- Is the user allowed to enroll for a certificate?
- Is there Issuance Requirements?
- Are information specified by CSR allowed in the certificate template settings?

4. Return Certificate

5. Client stores the certificate in the Windows Certificate store and uses it to perform actions allowed by the certificate(authentication, code signing, etc.)

Attack Path Preview - Certificate Template Abuse

- ENROLLEE_SUPPLIES_SUBJECT flag is enabled
- Excessive Permissions on Enrollment Agent Certificate
- EDITF_ATTRIBUTESUBJECTALTNAME2 setting on CA

```
CA Name                             : ca01.training_a.local\training_a-CA01-CA
Template Name                       : VulnCert
Schema Version                      : 2
Validity Period                     : 1 year
Renewal Period                      : 6 weeks
msPKI-Certificates-Name-Flag        : ENROLLEE_SUPPLIES_SUBJECT, SUBJECT_ALT_REQUIRE_DOMAIN_DNS
mspki-enrollment-flag               : NONE
Authorized Signatures Required      : 0
pkiextendedkeyusage                 : Client Authentication, KDC Authentication, Server Authentication, Smart Card Logon
mspki-certificate-application-policy  : Client Authentication, KDC Authentication, Server Authentication, Smart Card Logon
Permissions
  Enrollment Permissions
    Enrollment Rights           : NT AUTHORITY\Authenticated UsersS-1-5-11
                                  NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERSS-1-5-9
                                  training_a\Domain Admins      S-1-5-21-4048269214-1123341211-3658342892-512
                                  training_a\Domain Controllers S-1-5-21-4048269214-1123341211-3658342892-516
                                  training_a\Enterprise Admins  S-1-5-21-4048269214-1123341211-3658342892-519
                                  training_a\Enterprise Read-only Domain ControllersS-1-5-21-4048269214-1123341211-3658342892-498
    AutoEnrollment Rights       : NT AUTHORITY\ENTERPRISE DOMAIN CONTROLLERSS-1-5-9
                                  training_a\Domain Controllers S-1-5-21-4048269214-1123341211-3658342892-516
                                  training_a\Enterprise Read-only Domain ControllersS-1-5-21-4048269214-1123341211-3658342892-498
  Object Control Permissions
    Owner                       : training_a\Administrator      S-1-5-21-4048269214-1123341211-3658342892-500
    WriteOwner Principals       : training_a\Administrator      S-1-5-21-4048269214-1123341211-3658342892-500
                                  training_a\Domain Admins      S-1-5-21-4048269214-1123341211-3658342892-512
                                  training_a\Enterprise Admins  S-1-5-21-4048269214-1123341211-3658342892-519
    WriteDacl Principals        : training_a\Administrator      S-1-5-21-4048269214-1123341211-3658342892-500
                                  training_a\Domain Admins      S-1-5-21-4048269214-1123341211-3658342892-512
                                  training_a\Enterprise Admins  S-1-5-21-4048269214-1123341211-3658342892-519
    WriteProperty Principals    : training_a\Administrator      S-1-5-21-4048269214-1123341211-3658342892-500
                                  training_a\Domain Admins      S-1-5-21-4048269214-1123341211-3658342892-512
                                  training_a\Enterprise Admins  S-1-5-21-4048269214-1123341211-3658342892-519
```

Attack Path Preview - Certificate Authentication (PKINIT)

pc01.txone.corp.local
user

dc01.txone.corp.local
Domain Controller

1. AS-REQ (request TGT)

Pre-Authentication request sent to Domain Controller.

signedAuthPack = {Encrypt private key(Timestamp),…}
AS-REQ = {signedAuthPack, User Certificate…}

Authentication Server (AS)

1. Client Certificate Chain Validation
2. Decrypt signedAuthPack via a public key and verify the timestamp

2. AS-REP (receive TGT+Session Key)

The KDC verifies Authentication Data and replies with TGT

TGT=ticket-granting-ticket

Unconstrained Delegation Concept

Windows 10

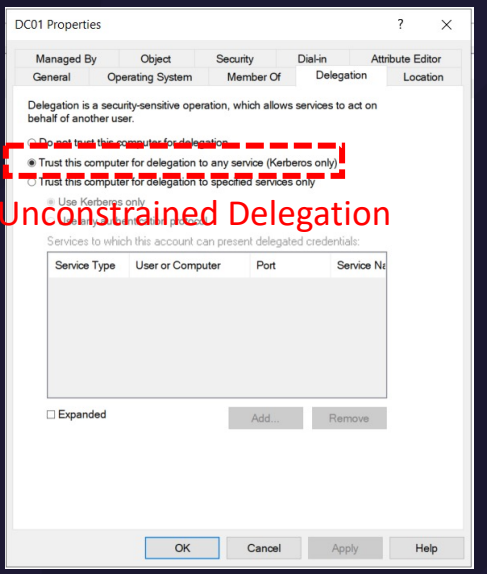
Microsoft Active Directory

Microsoft IIS

Microsoft SQL Server

1. AS-REQ (request TGT)
2. AS-REP (receive TGT)
3. TGS-REQ (TGT+ Request Ticket for IIS Service)
4. TGS-REP(Service Ticket for IIS Service +Session Key)
5. TGS-REQ(Request Forwarded Ticket for IIS Service)
6. TGS-REP(Receive Forwarded Ticket)
7. AP-REQ(present TGS for access and delegate his TGT)
8. TGS Request to SQL Service using the user's TGT
9. TGS for the user to SQL Service
10 .The IIS Service impersonates the user with the obtained TGS
11. Response

Forwarded TGT is cached for IIS to request service ticket for SQL Service



Unconstrained Delegation

txOne networks

Attack Path Preview - DCSync

- It is possible to have multiple domain controllers in the domain
- A change to one DC needs to be synced to another DC
  - This is called the domain replication
  - The DS-Replication-Get-Changes as well as DS-Replication-Get-Changes-All rights
- We can exploit replication protocol (user replication) to obtain user's password hashes

Attacker

1. Attacker MIMICS DC(MS-DRSR)

Discover DC and Mimic Replication

2. Request User Replication

Request Replication for credentials via GetNCChanges

3. DC provides password hashes

dc01.txone.corp.local
Domain Controller

Operation Security – Certificate Template Abuse

- The attacker exploits a vulnerable certificate template to obtain a certificate that can then be used for Kerberos authentication as a privileged account.
- Possible Indicator
  - The defender could monitor the enrollment event for “subject” or “subject alternative name” that has High privileged account specified
- OPSEC Actions
  - Instead of targeting high privileged accounts, such as administrator, for certificate template abuse, we can target the account that has specific rights we want
  - e.g.,
    1. Use exchange server account in certificate template abuse for DCSync attack
    2. Use a specific account that is the local admin to the machine we want to compromise

Operation Security – Certificate Authentication (PKINIT)

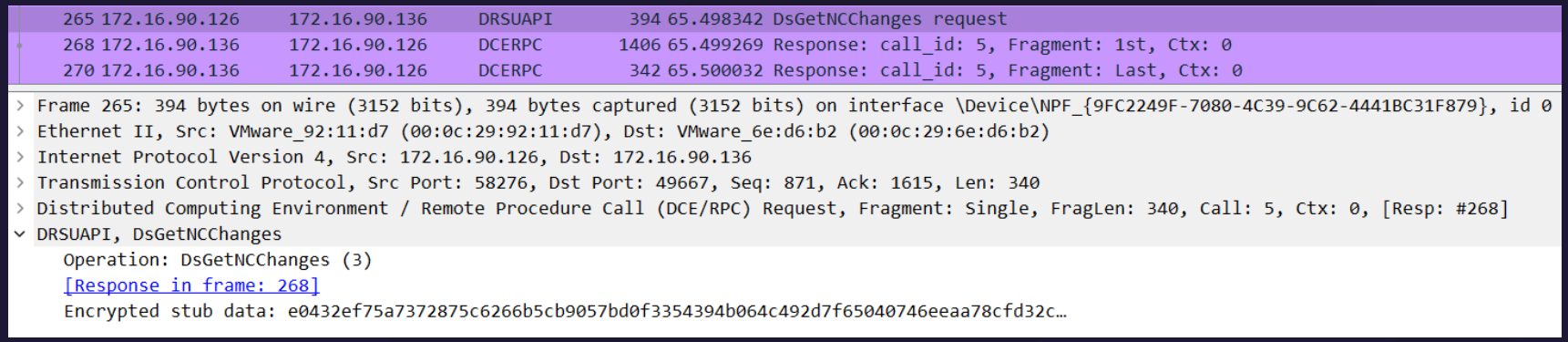
- The attacker uses the obtained certificate for Kerberos Authentication (PKINIT) and impersonates the target user to access a service
- Possible Indicator
  - PKINIT can be alerted by monitor the “certificate information” in TGT request event (4768)
- OPSEC Risk
  - PKINIT may be rarely happening in the target environment, using the certificate for Kerberos authentication may easily be stood out

Operation Security - Unconstrained Delegation

- Attacker abuses the unconstrained delegation to dump the TGTs of users who access the computer that has the setting configured
  - In the scenario of cross forest trust, default unconstrained delegation on DC is often leveraged with coerce authentication, such as printer bug, to obtain the TGT of foreign DC
- Possible Indicator
  - For ticket dump, **LSASS related activities could be heavily monitored on DC** that is defined as Tier-0 asset e.g., LSA handle request mentioned earlier
- OPSEC
  - For cross forest trust scenario, we can configure unconstrained delegation settings on a regular domain computer compromised to avoid the heavy monitor for dump tickets on DC

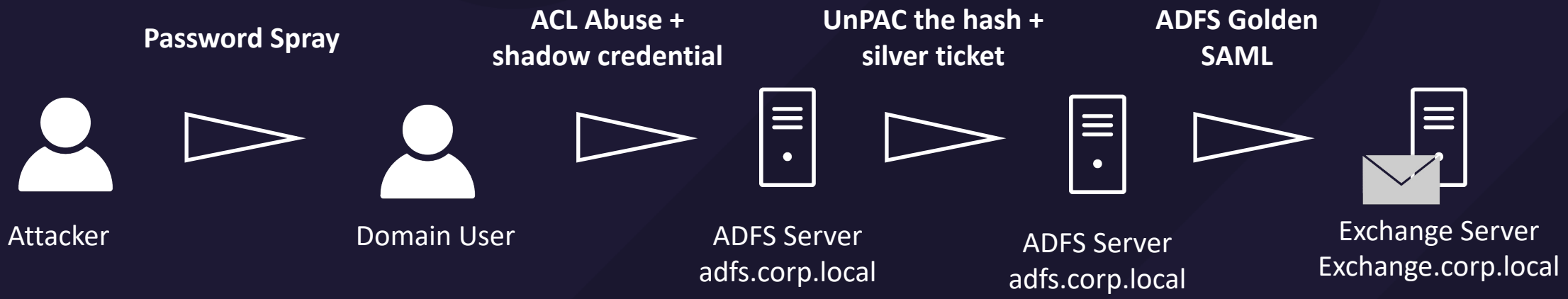
Operation Security - DCSync

- Attacker leverage DCSync to dump the password hash of domain accounts
- Possible Indicator
  - Monitoring traffic for DsGetNCChanges requests from not whitelisted machine
- OPSEC Actions
  - We can attempt to find a machine that **“might” be in the whitelist** such as AAD connect or just having a shell on a DC to avoid traffic monitoring



Attack Path III Overview

From Password Spray to Abuse ADFS Golden SAML Token



Attack Path Preview - Password Spray

- To obtain more access to accounts, adversaries may leverage the services that will check for the validity of credentials to brute force the password
  - Password Spraying is a type of brute force attack where the attacker uses a **single common password against multiple accounts to avoid lockouts**

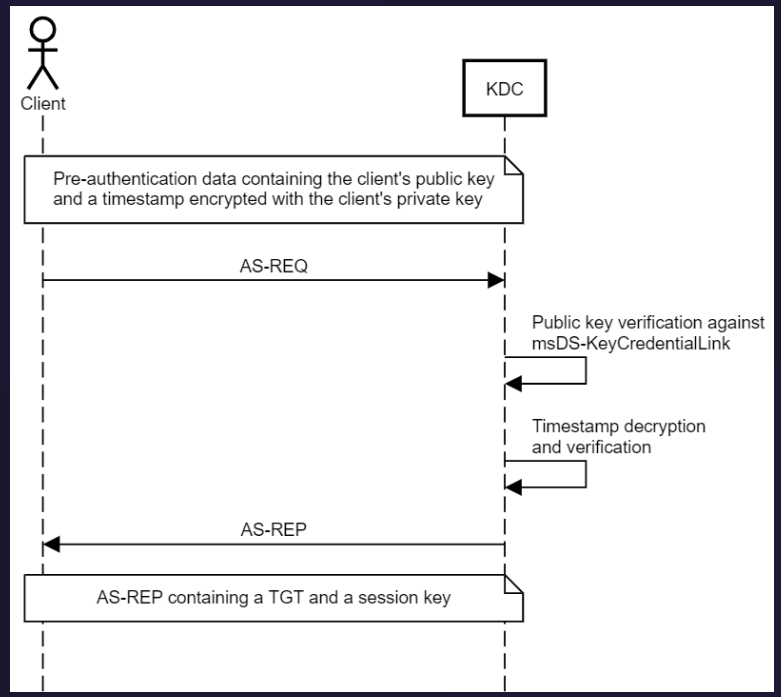
Attack Path Preview - ACL Abuse

- In Active Directory, every resource is managed and stored as domain object and each domain object can grants certain security principals some rights (ACL) such as write property and full control
  - For different domain object type and different right, there are various way of attacks implementation
  - e.g.,
    - Force change the password to compromise a domain account
    - Add compromised account to the member of DC groups for DCSync attack

Permission	Equivalent to	ACL abuse
writeOwner	RIGHT_WRITE_OWNER	Change the object's owner to grant ourself for writeDacl rights
writeDacl	RIGHT_WRITE_DAC	Escalate to genericAll grant ourself specific rights e.g. change passwd
genericWrite	RIGHT_READ_CONTROL RIGHT_DS_WRITE_PROPERTY RIGHT_DS_WRITE_PROPERTY_EXTENDED	From attacker's perspective it is the same as genericAll
genericAll	RIGHT_DELETE RIGHT_READ_CONTROL RIGHT_WRITE_DAC …	Modify object attribute or add extended right to exploit

Attack Path Preview - Shadow Credential

- Certificate (Shadow Credential) used for authenticating an account can be configured in “msDS-KeyCredentialLink” attribute to enable PKINIT without AD CS installed
  - The attacker often abuses the ACL to configure the certificate on an account object to ask TGT and impersonate this account



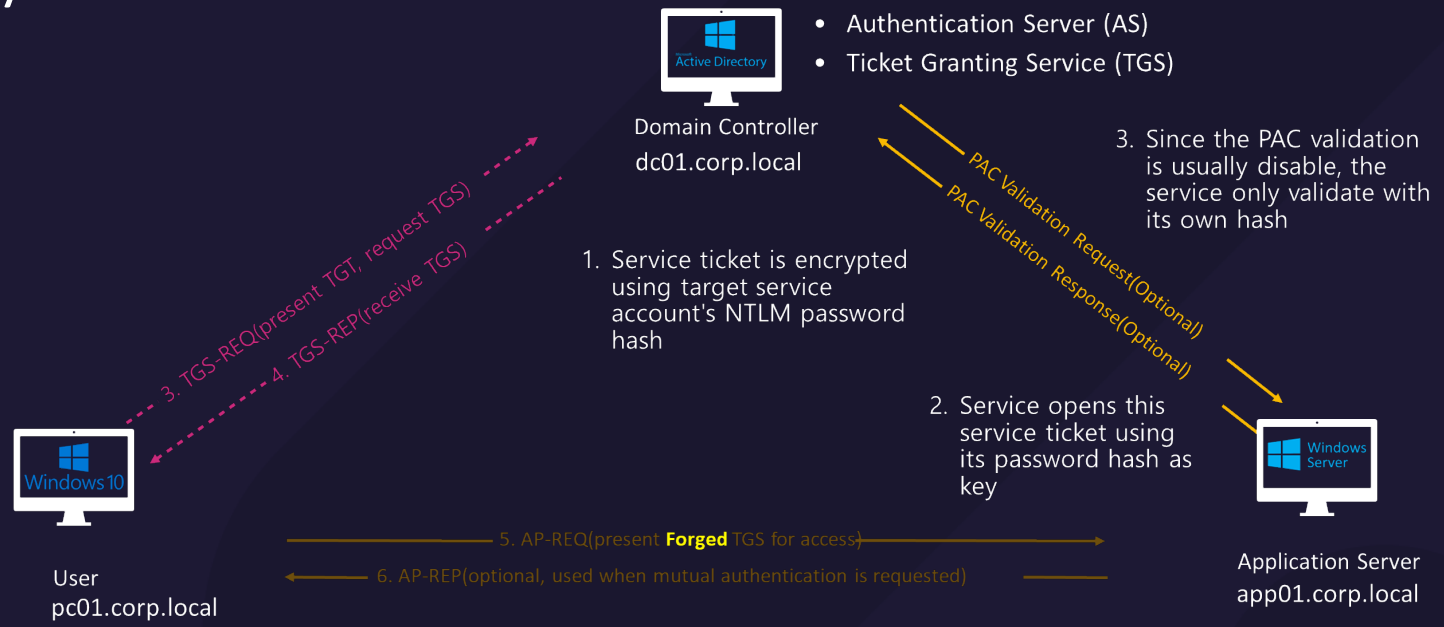
Attack Path Preview - UnPAC the hash

- When a TGT is requested through PKINIT, the RC4 password hash of the user is included in the PAC of that ticket
  - The attacker can use this TGT to request a U2U service ticket and obtain the RC4 password hash of the target user by decrypting the service ticket PAC with the session key

```
CredentialInfo         :
  Version              : 0
  EncryptionType       : rc4_hmac
  CredentialData       :
    CredentialCount    : 1
    NTLM               : A6FF4D034DBCF0C1DA62F86D34D45754
```

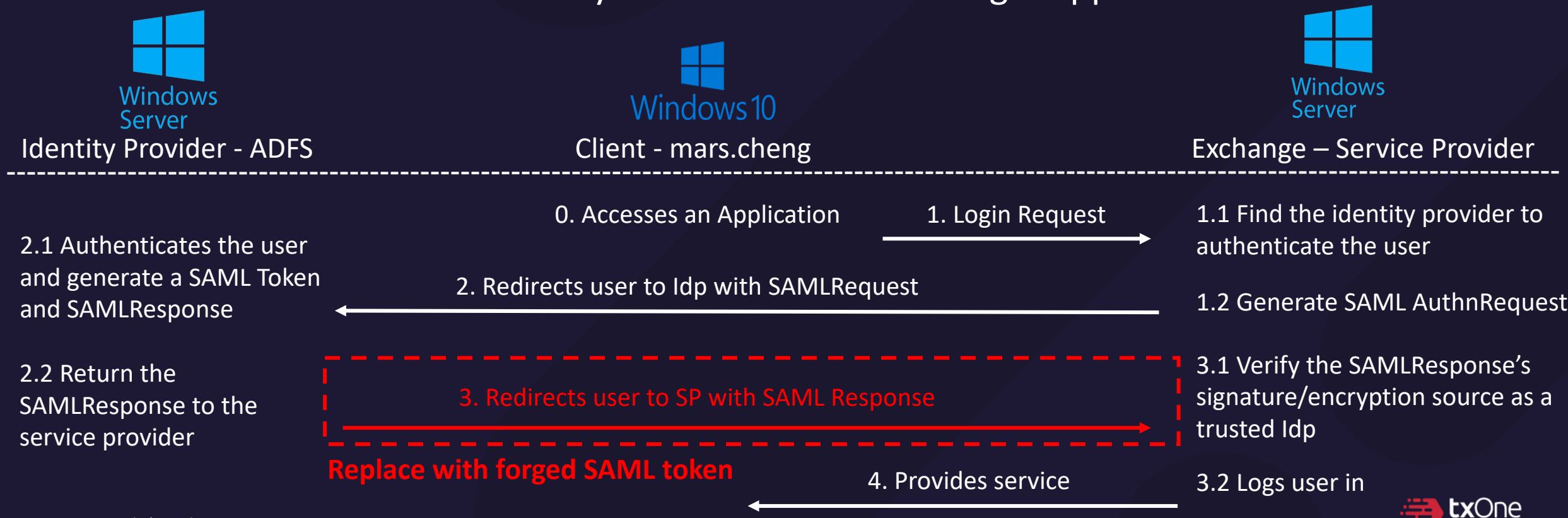
Attack Path Preview - Silver ticket

- Attacker who has the password hash of a target service account may forge the service ticket with arbitrary principal that can be specified for accessing the service
  - UnPAC the hash can be leveraged to forge the silver ticket and compromise a computer system



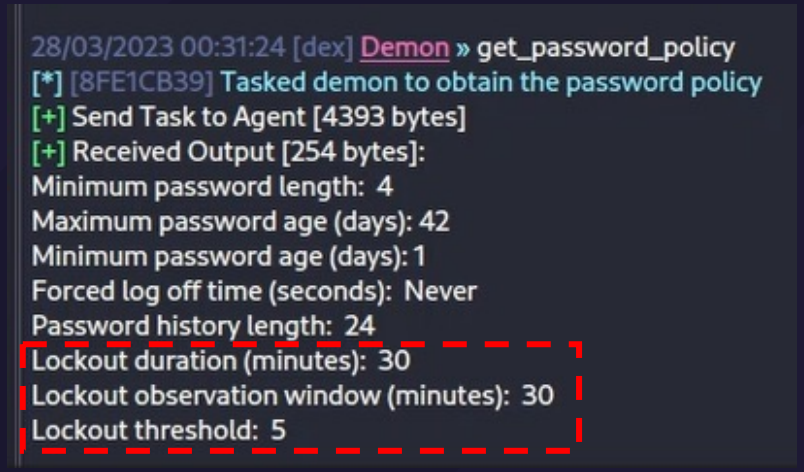
Attack Path Preview - ADFS Golden SAML

- Attackers forged the token to access any application that supports SAML authentication enabled by ADFS
  - The SAML token can be any user we desired for target application



Operation Security - Password Spray

- The attacker attempts to brute force each password against a list of domain users
- Possible Indicator
  - The high volume of logon failures or account disabled in a short period of time
- OPSEC Action
  - Increase the time between brute force attempts and track the counter of bad password attempts for not exceeding the threshold that will disable the account



Operation Security - ACL Abuse for Shadow Credential

- Attacker leverage ACL abuse to configure a certificate (shadow credential) on a domain account object for Kerberos Authentication (PKINIT) and impersonates the target user to access a service
- Possible Indicator
  - Shadow Credential can be alerted by monitoring write event of “msDS-KeyCredentialLink” attribute for account object
- OPSEC Risk
  - Depending on the target environment, configure certificate via writing “msDS-KeyCredentialLink” attribute may easily be stood out

Operation Security - UnPAC the Hash for Silver Ticket

- After the shadow credential is configured, the attacker obtains the password hash via UnPAC the hash and leverage it to forge the silver ticket
- Possible Indicator
  - Since the password hash obtained from UnPAC the hash technique is RC4, the silver ticket forged will have an encryption type limited to RC4, using RC4 will be alerted
- OPSEC Action
  - Instead of leveraging UnPAC the hash + silver ticket to compromise computer system after the shadow credential is configured, an alternative option can be using S4U2Self

Operation Security - UnPAC the Hash for Silver Ticket

- After shadow credential is configured for a computer account, we can ask the TGT and leverage S4U2Self to impersonate a privileged user for access
  - The encryption type for ticket from S4U2Self will align to the baseline since it is

```
[*] Action: Ask TGT

[*] Using PKINIT with etype aes256_cts_hmac_sha1 and subject: CN=ADFS01$
[*] Building AS-REQ (w/ PKINIT preauth) for: 'corp.local\ADFS01$'
[*] Using domain controller: 172.16.90.136:88
[+] TGT request successful!
[*] base64(ticket.kirbi):

      doIFzDCCBcigAwIBBaEDAgEWooIE2DCCBNRhggTQMIIEzKADAgEFoQwbCkNPUlAuTE9DQU
FkvdtMpMr9yWJXsMaOhHp3ey57c7EcOekpv96nU2ZGZ5ypWWxV1cz3Z0AmDFM9O9BXocpZbiDMcc
frV7nsUIh+sOyT8YadMyNUAt//n/6TdnvVwPduXQmL1aGOcrATrnuTOV0/JFYtCQqgjgqrcpeNxw
dL2RVIEZJ8Y4+R1dCbP3YjWPI4cvycZ5SxgqJJcc1OwH0Gv9D22jXy+0qq8ctMPpo6AAQFPFWZAJ
```

1

PKINIT

```
[*] Action: S4U

[*] Building S4U2self request for: 'ADFS01$@CORP.LOCAL'
[*] Using domain controller: DC01.corp.local (172.16.90.136)
[*] Sending S4U2self request to 172.16.90.136:88
[+] S4U2self success!
[*] Substituting alternative service name 'cifs/a
[*] Got a TGS for 'administrator' to 'cifs@CORP.L
[*] base64(ticket.kirbi):
```

s4u2self

```
[*] Action: Describe Ticket

ServiceName              :  cifs/adfs01.corp.local
ServiceRealm             :  CORP.LOCAL
UserName                 :  administrator
UserRealm                :  CORP.LOCAL
StartTime                :  4/11/2023 10:51:01 PM
EndTime                  :  4/12/2023 8:48:55 AM
RenewTill                :  4/18/2023 10:48:55 PM
Flags                    :  name_canonicalize, pre_authent, renewable
KeyType                  :  aes256_cts_hmac_sha1
Base64(key)              :  5bUmlsa65whzgx9iu/8x4GAJTxwPnxCUG7FDAKFg3Fw=
```

2

```
[+] Ticket successfully imported!
PS C:\Users\mars.cheng\Desktop> ls \\adfs01.corp.local\c$


    Directory: \\adfs01.corp.local\c$


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         3/27/2023   4:27 AM                ExchangeSetupLogs
d-----         9/15/2018  12:19 AM                PerfLogs
d-r---         8/29/2021   1:46 PM                Program Files
d-----         9/15/2018   2:08 AM                Program Files (x86)
d-r---         3/23/2023  10:35 PM                Users
d-----         3/27/2023   9:08 PM                Windows
```

3

Access target

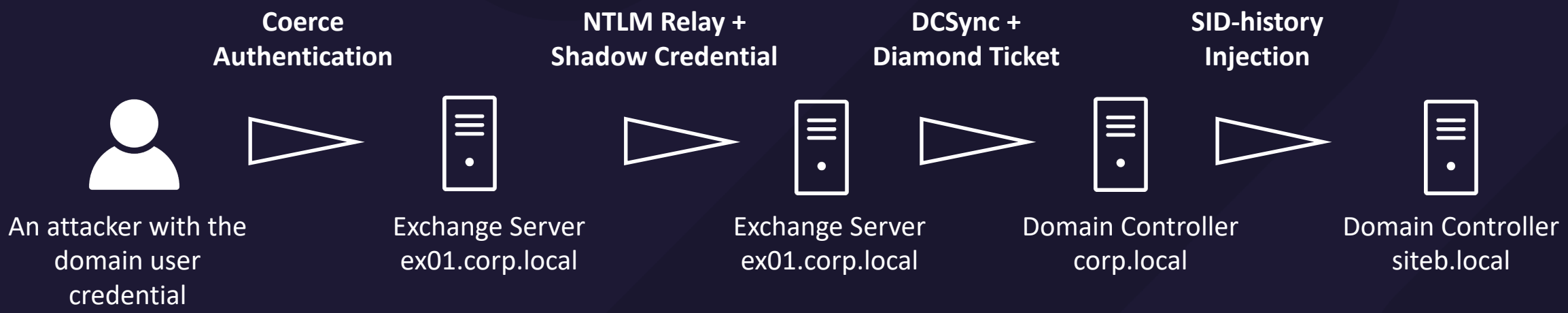
txOne networks

Operation Security - ADFS Golden SAML

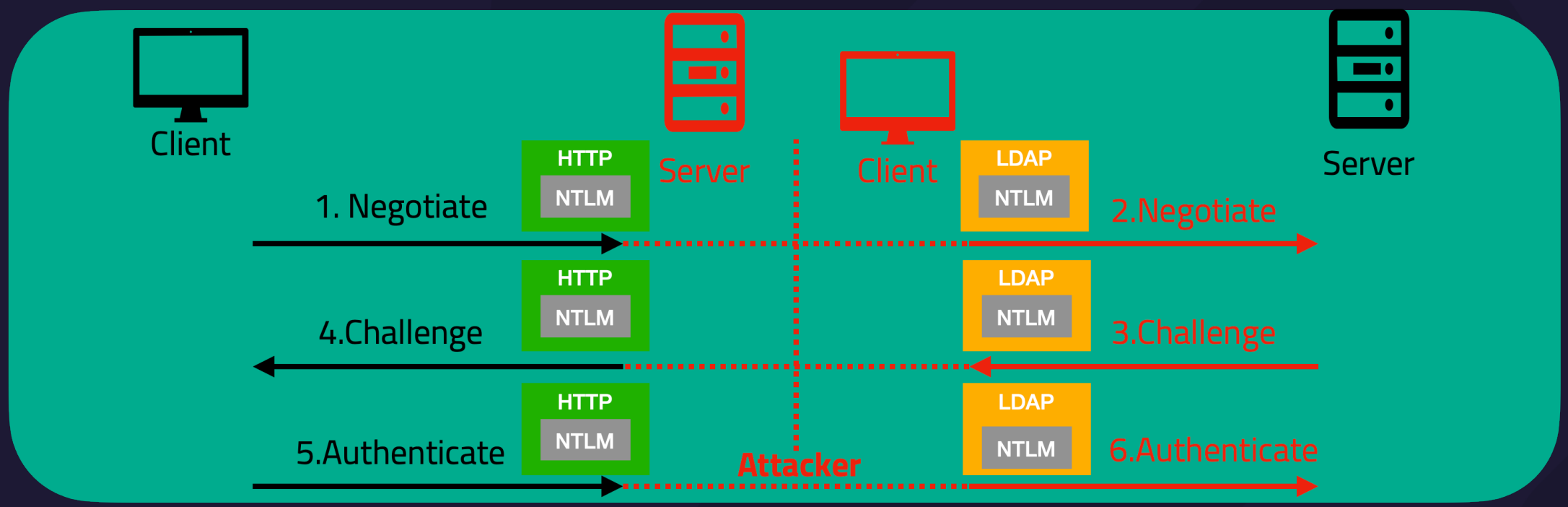
- The attacker dump credential, DKM key, and configs, from the ADFS server to forge the SAML token and access the application
- Possible Indicator
  - Places, such as DB and registry key, that save the credential could **be heavily monitored on ADFS Server** which is defined as a Tier-0 asset
- OPSEC Action
  - Instead of dumping the credential locally on the ADFS server, methods exist to dump them remotely
    - Leverage LDAP or DCSync for DKM key
    - Leverage AD FS replication service for configurations

Attack Path IV Overview

Leverage Printer Bug for Shadow Credentials



Attack Path Preview - NTLM Relay + Shadow Credential



7. Set Certificate to msDS-KeyCredentialLink attribute

8. Send Certificate to request TGT via PKINIT

9. Response TGT to Client

Shadow Credential

Attack Path Preview - Diamond Ticket

- The attacker uses **krbtgt password hash** to **add** a privileged group to the **membership attribute of PAC** to a valid TGT requested or stolen
  - Instead of forging a new ticket like Golden/Silver ticket, a Diamond ticket evades the common detection indicator (such as 10 years valid time or non-baseline encryption type) for a forged ticket
- Attack Procedure
  - Request a TGT for an arbitrary user
  - Decrypt the PAC in TGT using the krbtgt password hash
  - **Modify the desired field in PAC for privileged access**
  - Encrypt the PAC again using the krbtgt password hash

Attack Path Preview - SID-history Injection

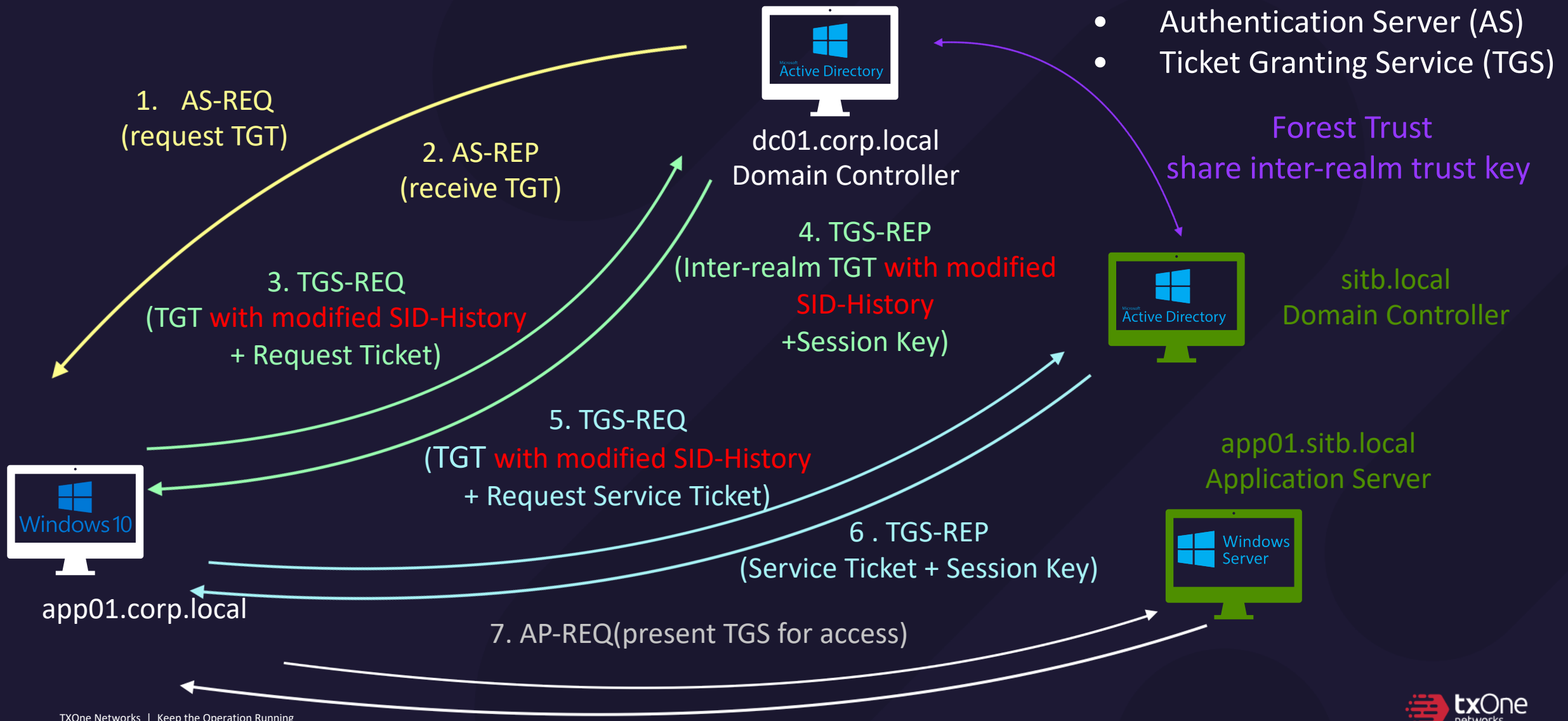
- The SID-History injection is done by adding SIDs to the SID-History field in the PAC of a TGT by using the krbtgt password hash
  - Allow an account to hold additional identity
    - S-1-5-21-<Domain>-500 Administrator
    - S-1-5-21-<Domain>-512 Domain Admins
    - S-1-5-21-<Domain>-516 Domain Controllers
- By adding the SID of the foreign domain administrator, the attacker will have admin privilege for the foreign domain controller

SID-History Injection

```
ServiceName           :  krbtgt/CORP.LOCAL
 ServiceRealm         :  CORP.LOCAL
 UserName             :  Administrator
 UserRealm            :  CORP.LOCAL
 StartTime            :  4/23/2022 9:16:33 PM
 EndTime              :  4/24/2022 7:16:33 AM
 RenewTill            :  4/30/2022 9:16:33 PM
 Flags                :  name_canonicalize, pre_authent, initial, renewable, forwardable
 KeyType              :  aes256_cts_hmac_sha1
 Decrypted PAC        :
  LogonInfo           :
   LogonTime          : 4/23/2022 8:49:32 PM
   PasswordLastSet    : 4/22/2022 10:00:23 PM
   PasswordCanChange  : 4/23/2022 10:00:23 PM
   EffectiveName      : Administrator
   Groups             : 512,513,520,518,519
   UserFlags          : (32) EXTRA_SIDS
   LogonDomainId      : S-1-5-21-4048269214-1123341211-3658342892
   UserAccountControl : (528) NORMAL_ACCOUNT, DONT_EXPIRE_PASSWORD
   ExtraSIDCount      : 1
   ExtraSIDs          : S-1-18-1
   ResourceGroupCount : 0
  UpnDns              :
   DNS Domain Name    : CORP.LOCAL
   UPN                : Administrator@corp.local
   Flags              : 1
  ServerChecksum      :
   Signature Type     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
   Signature          : CC4DFF201F1B89A3BE1EE560 (VALID)
  KDCChecksum         :
   Signature Type     : KERB_CHECKSUM_HMAC_SHA1_96_AES256
   Signature          : 6F370E92E7B86AD46CA93215 (VALID)
```

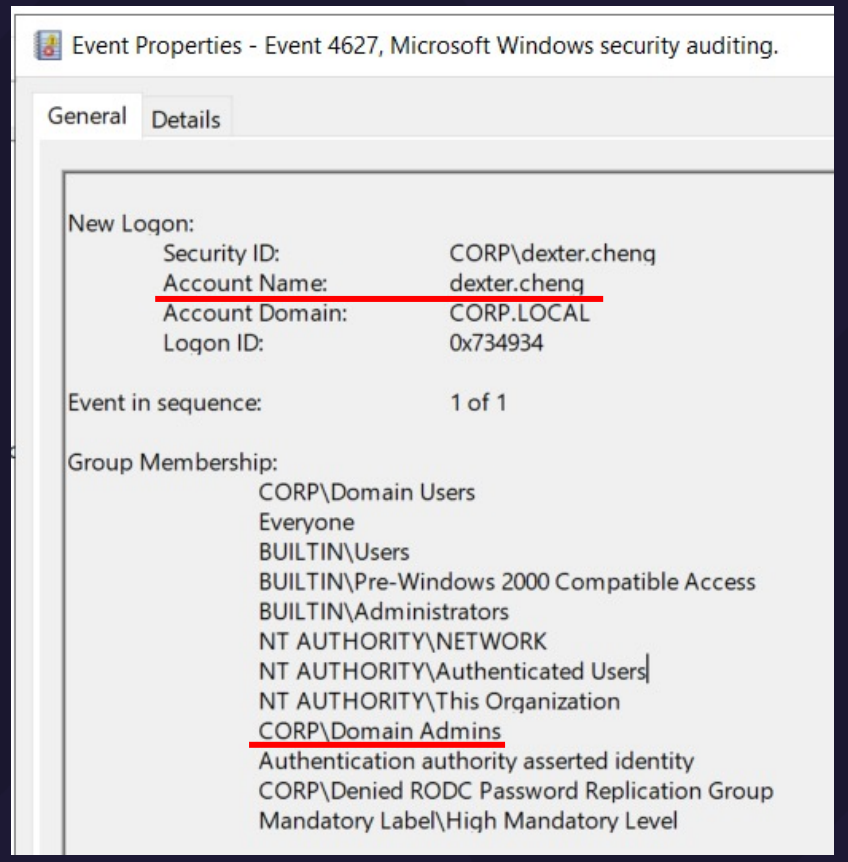
```
ServiceName           :  krbtgt/corp.local
 ServiceRealm         :  CORP.LOCAL
 UserName             :  administrator
 UserRealm            :  CORP.LOCAL
 StartTime            :  4/24/2022 04:49:14
 EndTime              :  4/24/2022 14:49:14
 RenewTill            :  5/1/2022 04:49:14
 Flags                :  pre_authent, initial, renewable, forwardable
 KeyType              :  rc4_hmac
 Decrypted PAC        :
  LogonInfo           :
   LogonTime          : 4/24/2022 04:49:14
   PasswordLastSet    :
   PasswordCanChange  :
   EffectiveName      : administrator
   Groups             : 520,512,513,519,518
   UserFlags          : (32) EXTRA_SIDS
   LogonDomainId      : S-1-5-21-4048269214-1123341211-3658342892
   UserAccountControl : (16) NORMAL_ACCOUNT
   ExtraSIDCount      : 1
   ExtraSIDs          : S-1-5-21-2363771286-3165302148-3346416472-500
   ResourceGroupCount : 0
  UpnDns              :
   DNS Domain Name    : CORP.LOCAL
   UPN                : administrator@corp.local
   Flags              : 0
  ServerChecksum      :
   Signature Type     : KERB_CHECKSUM_HMAC_MD5
   Signature          : 529D718A03362A7D2BD59415238C020F (VALID)
  KDCChecksum         :
   Signature Type     : KERB_CHECKSUM_HMAC_MD5
   Signature          : 658F7F1CF6848ADBCA318C898E762C7A (VALID)
```

Kerberos Authentication on Forest Trust



Operation Security - Diamond Ticket

- Attacker have elevated access by **modifying** the membership of PAC for a TGT
- Possible Indicator
  - PAC's value and actual AD relationship can be used as detection indicator
    e.g., PAC in a TGT defines user in Administrators group but in fact it is not
- OPSEC Action
  - We can also use sapphire ticket which leverage S4U2self and U2U to obtain target user ticket for corresponding PAC value



Takeaways

txOne networks

Takeaways

- Abuse Primitives are still the main trend for attacking AD since they are hard to be completely prevented
- There will be more detection mechanism deployed gradually for Active Directory abuse primitives
- Operation Security(OPSEC) should be considered before an action is performed

Thank you for your attention

Keep the operation running!

Mars Cheng, mars_cheng@txone.com, @marscheng_
Dexter Chen, dexter_chen@txone.com